



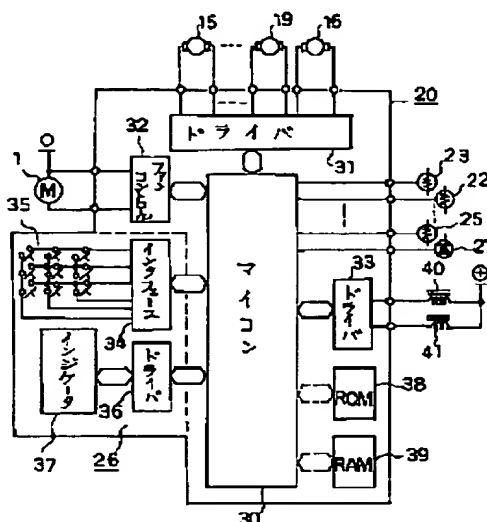
PATENT ABSTRACTS OF JAPAN

(11) Publication number: **07093137 A**(43) Date of publication of application: **07.04.95**(51) Int. Cl. **G06F 9/06**(21) Application number: **06061072**(22) Date of filing: **30.03.94**(62) Division of application: **01240082**(71) Applicant: **HITACHI LTD**(72) Inventor: **KANEHATA YASUO****(54) AUTOMOBILE CONTROLLER AND PROGRAM DEVELOPING DEVICE****(57) Abstract:**

PURPOSE: To reduce man-hour for developing an automobile controller or man power for development and to reduce the manufacture cost by describing a control program in any high-class language and storing the translated result of the control program into a machine word in the built-in memory of a one-chip microcomputer to be used for the automobile controller.

CONSTITUTION: The automobile controller 20 is provided with a one-chip microcomputer 30 which incorporates the memory for storing the control program prepared in the high class language in the form translated into the machine word. The one-chip microcomputer 30 inside the controller 20 incorporates a processor, ROM, RAM, I/O port and A/D converter or the like. This, since the control program is described in the high class language so that the high class language can facilitate the comprehension of contents and simplify programming, the control can be easily prepared and changed even for a program other than the specified program and further, since the number of lines in the program of assembly is decreased to be one divided by several numbers or one divided by several tens in comparison with that of a language, the program can be easily developed, maintained and certified.

COPYRIGHT: (C)1995,JPO



**AUTOMOBILE CONTROLLER AND PROGRAM DEVELOPING DEVICE**

Publication Number: 07-093137 (JP 7093137 A), April 07 1995

Inventors:

- KANEHATA YASUO

Applicants

- HITACHI LTD (A Japanese Company or Corporation), JP (Japan)

Application Number: 06-061072 (JP 9461072), March 30, 1994

International Class (IPC Edition 6):

- G06F-009/06

JAPIO Class:

- 45.1 (INFORMATION PROCESSING--- Arithmetic Sequence Units)
- 24.2 (CHEMICAL ENGINEERING--- Heating & Cooling)
- 26.2 (TRANSPORTATION--- Motor Vehicles)

JAPIO Keywords:

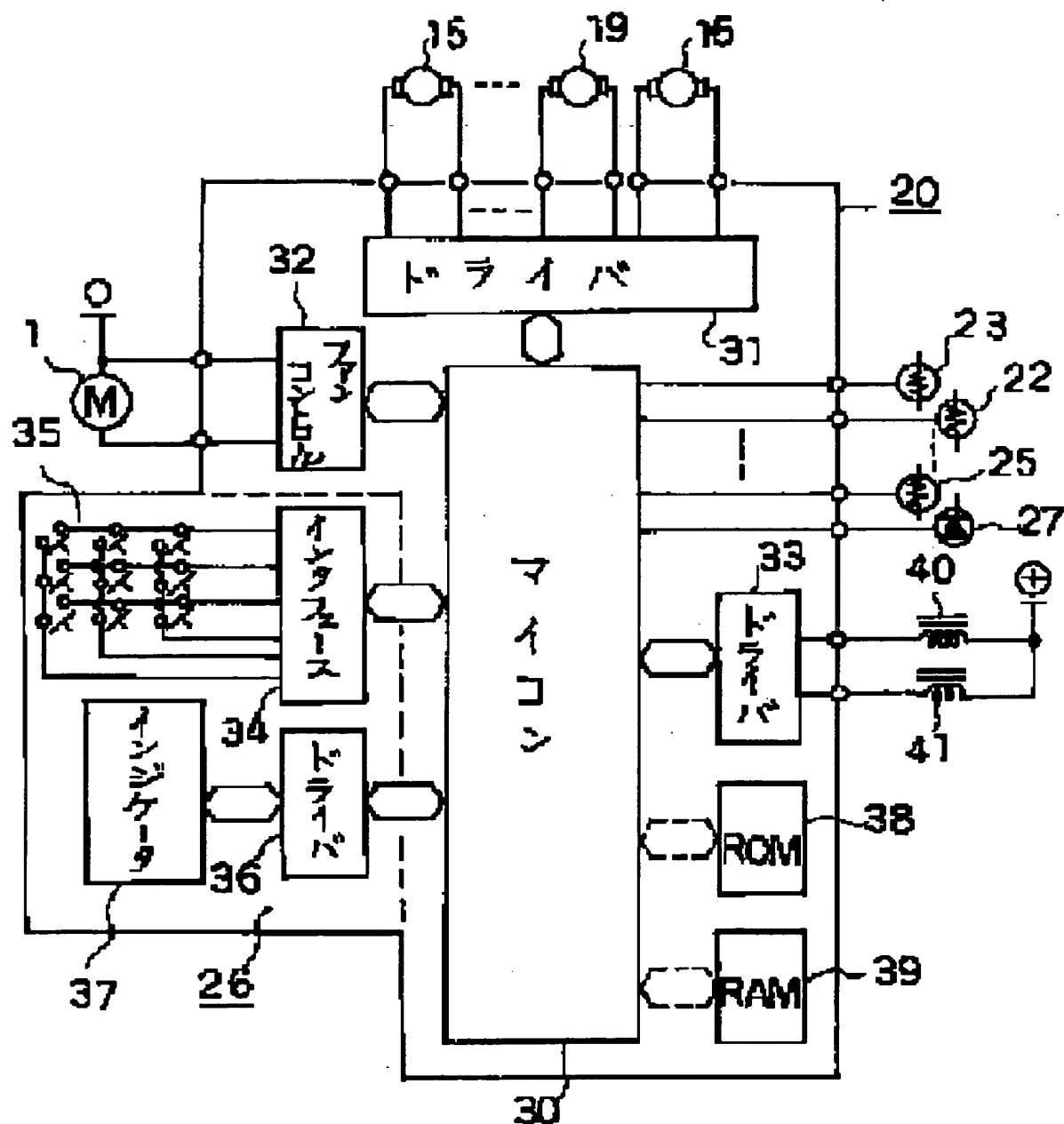
- R131 (INFORMATION PROCESSING--- Microcomputers & Microprocessors)

Abstract:

PURPOSE: To reduce man-hour for developing an automobile controller or man power for development and to reduce the manufacture cost by describing a control program in any high-class language and storing the translated result of the control program into a machine word in the built-in memory of a one-chip microcomputer to be used for the automobile controller.

CONSTITUTION: The automobile controller 20 is provided with a one-chip microcomputer 30 which incorporates the memory for storing the control program prepared in the high class language in the form translated into the machine word. The one-chip microcomputer 30 inside the controller 20 incorporates a processor, ROM, RAM, I/O port and A/D converter or the like. This, since the control program is described in the high class language so that the high class language can facilitate the comprehension of contents and simplify programming, the control can be easily prepared and changed even for a program other than the specified program and further, since the number of lines in the program of assembly is decreased to be one divided by several numbers or one divided by several tens in comparison with that of a language, the program can be easily developed, maintained and certified.

See also
ISR
+
IPER



JAPIO

© 2000 Japan Patent Information Organization. All rights reserved.

Dialog® File Number 347 Accession Number 4800537

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-93137

(43) 公開日 平成7年(1995)4月7日

(51) Int.Cl.⁸

G 0 6 F 9/06

識別記号

庁内整理番号

9367-5B

F I

技術表示箇所

審査請求 有 請求項の数2 O L (全 11 頁)

(21) 出願番号 特願平6-61072
(62) 分割の表示 特願平1-240082の分割
(22) 出願日 平成1年(1989)9月18日

(71) 出願人 000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目6番地
(72) 発明者 鹿子橋 庸雄
茨城県勝田市大字高場2520番地 株式会社
日立製作所佐和工場内
(74) 代理人 弁理士 秋本 正実

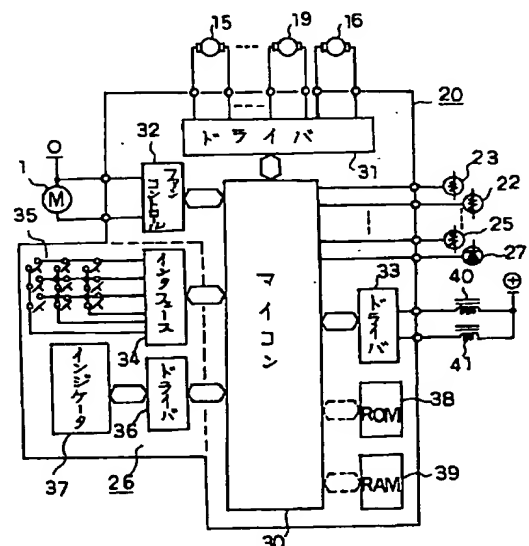
(54) 【発明の名称】 自動車制御装置とプログラム開発装置

(57) 【要約】

【目的】 自動車制御装置の開発工数や開発マンパワーの低減を図り製造コストを低減させる。

【構成】 各種センサから取り込んだ検出信号に所定の制御プログラムによる演算処理を施し、処理結果を制御信号として制御対象に出力する自動車制御装置において、高級言語で作成された制御プログラムを機械語に翻訳された形で格納するメモリを内蔵した1チップマイコンを自動車制御装置に設ける。高級言語は、アセンブリ言語に比較して扱えるプログラマも多く、扱い易い為、プログラムの開発工数や人数が減少する。

[図 3]



【特許請求の範囲】

【請求項1】 各種センサから取り込んだ検出信号に所定の制御プログラムによる演算処理を施し、処理結果を制御信号として制御対象に出力する自動車制御装置において、高級言語で作成された前記制御プログラムを機械語に翻訳された形で格納するメモリを内蔵した1チップマイコンを備えることを特徴とする自動車制御装置。

【請求項2】 自動車用DC電源で動作し、請求項1記載の制御プログラムを開発する車載用のプログラム開発装置において、制御用の1チップマイコンを搭載し車輛機器を制御する制御装置のバスとコネクタを介して接続されるバスと、該バスに接続され高級言語で作成された制御プログラムを機械語に翻訳して実行するマイコンと、前記1チップマイコンのOSと共通仕様のOSであって車輛走行中に前記制御装置に接続された各種センサからの検出信号を取り込んで前記制御プログラムにより演算処理を行い前記車輛機器を制御している最中に前記制御プログラムを修正し機械語に再翻訳して前記マイコンに実行させるリアルタイムマルチタスクOSとを備えることを特徴とするプログラム開発装置。

【発明の詳細な説明】**【0001】**

【産業上の利用分野】 本発明は自動車制御装置とプログラム開発装置に関する。

【0002】

【従来の技術】 自動車用のマイクロコンピュータ（以下マイコンと略称）は、エンジンの点火・燃料制御、エアコン制御、トランスミッション制御、アクティブサスペンション制御、あるいはブレーキ制御等の各種の自動車制御装置に用いられ、しかも各制御装置ごとにそれぞれ専用の1チップマイコンを用意するのが普通である。このような1チップマイコンとしては、例えば特開昭63-11169号に示されたエアコン制御用のものがあり、制御用に設計された電子回路に合わせてその制御プログラムが作られている。このような自動車用の1チップマイコンの制御プログラムは、通常アセンブリ言語（または機械語）で記述されているが、その理由は主に次の2つである。

(1) メモリ（ROM、RAM）容量の制約がきびしい。

(2) 早い処理速度が要求される。

【0003】 ここで(1)について考えると、自動車部品の低コスト化が非常にきびしいため、マイコンとしては価格的に有利な4ビットまたは8ビットの1チップマイコンが用いられてきた。ところが1チップマイコンはその集積度の制約からROM、RAMともわずかな容量しか内蔵できず、これらの中で必要な制御仕様を満足するためには、プログラムが最もコンパクトであり、RAMも有効に活用できるアセンブリ言語でプログラムを記述せざるを得なかった。メモリ容量の制約から逃れるために、外部に別チップのROM、RAMを増設すると、そ

の増設ROM、RAM以外に、アドレスデータ用のICや各チップ間接続のためのプリント基板上の配線パターン等にスペースが必要となり、部品の増加、制御回路の大形化、大幅な価格上昇等を引き起こしてしまう。

【0004】 一方、(2)の条件について考えてみると、従来のマイコンは処理速度が比較的遅いため、プログラムのステップ数が増加するのに従って、メインプログラム1周に要する時間が長くなり、制御系の応答性が次第に悪化する。割込処理プログラムの構成方式にも大きく影響を受けるが、実際にメインプログラムの1周に0.5秒程度の時間を要するものもめずらしくない。従って、制御システムの応答性を向上させるためにも、プログラムの1周に要するステップ数は極力少ないことが要求され、この条件からもプログラムが最もコンパクトで、早い処理速度が得られるアセンブリ言語を使用せざるを得なかった。

【0005】

【発明が解決しようとする課題】 近年の集積回路の製造技術の進歩により、1チップマイコンに内蔵されるRAM、ROMの容量は増大する傾向にある。しかし、家庭用電気製品等に使用されるマイコンと異なり、自動車制御装置として使用される1チップマイコンは、自動車のエレクトロニクス化の進展により1つの1チップマイコンが処理するタスク数は増加の一途を辿り、従って、容量の増大したメモリに何種類もの制御プログラムを搭載しなければならない。このため、容量の増大したメモリに対しても無駄なく制御プログラムを格納したいという要請がある。更に、自動車制御装置は、数ms、数十msという決まった時間内に必ず処理を終了して演算結果を出力しなければならないタスクを周期的に実行しており、高速処理に対する要請もある。

【0006】 アセンブリ言語は、実行されるプログラムがコンパクトになり、メモリ容量の低減と処理の高速化が行えるという利点を持つため、自動車制御装置にとって上記の両方の要請に同時に答えることができる。しかしその反面、アセンブリ言語の作成は、ハードウェア動作を1ステップずつ記述するため、プログラミング作業が複雑となり、プログラムの意味が分かり難く、プログラムリストの行数が大きくなる等の欠点がある。このため、次の問題点がある。

【0007】 (1) アセンブリ言語を扱える数少ないプログラムでないと制御プログラムの作成・変更が困難で、プログラムの検証にも長時間を要する。

【0008】 (2) アセンブリ言語は異なるアーキテクチャのマイコン間では全く互換性を持たないから、マイコン機種の変更時には全てのプログラムを作り直さなければならず、またプログラムの修正・変更も使用マイコンと同じアーキテクチャの設備を必要とし、実験室や屋外走行実験中の即座のプログラム変更を困難とする。

【0009】 (3) 動作確認の実験時などにマイコン内部

のデータ記録やその外部への表示機能等を組込むことは、メモリ容量の制約から困難であり、制御応答性の検証や仕様の決定作業が難しい。

【0010】(4) プログラム開発装置と最終のマイコン製品とは、一般にアーキテクチャが異なるため、開発装置上で動作検証を行ったプログラムを最終製品に移す時点でプログラムの一部変更（特に入出力処理に関する部分の変更）が必要となり、作業性とプログラムの信頼性が低下する。

【0011】本発明の目的は、アセンブリ言語を扱うように訓練されていない一般のプログラマでも制御プログラムの作成・変更及び検証が容易に行え、マイコン機種を変更しても制御プログラムを変更する必要がなく、実験場所でのプログラム変更を容易とした自動車制御装置とそのプログラム開発装置を提供するにある。

【0012】

【課題を解決するための手段】上記の目的は、制御プログラムを高級言語により記述し、その制御プログラムを機械語に翻訳したものを自動車制御装置に用いる1チップマイコンの内蔵メモリに格納することで、達成される。

【0013】また、プログラムの開発装置と制御プログラムのオペレーティングシステム（以下OSと略称）を共通の仕様とし、開発装置の上で制御プログラムを走らせて対象機器を直接制御できるようにすることで、達成される。

【0014】

【作用】アセンブリ言語ではなく、高級言語により制御プログラムを記述すれば、高級言語は内容の理解が容易でプログラミングも簡単になるから、特定のプログラマ以外でも制御プログラムの作成・変更が容易に行え、かつアセンブリ言語に対しプログラム行数が数分の一から数十分の一に減少するため、プログラムの開発、保守、検証が容易となる。

【0015】プログラム開発装置のOSとマイコンのOSを共通仕様とすることにより、制御プログラムの動作環境が開発装置上とマイコン上で同じとなり、開発装置上で検証された制御プログラムは何の変更もなしに制御装置へ移せる。従って、制御装置に合わせてプログラムの一部を修正する作業が不要となり、作業期間の短縮、新たなミスの発生の防止に役立つ。また、プログラム開発装置内のプログラムにより、実際の対象機器の制御を直接行えるから、プログラム開発装置の大容量RAMや他の記録装置及び外部機器との入出力機能が活用でき、これらの機能を利用して任意データの記録や分析、及び車両上等での実験中のプログラム変更等が容易に行える。

【0016】さらに、制御プログラムの入出力処理にOSへのシステムコールを利用することによって、開発装置及びマイコン機種間のアドレスの相違や入出力部分の

処理手順の相違は、全てOSの内部に覆いかくされてしまうので、制御プログラム上からは見えなくなり、マイコン機種、使用部品、開発装置の構成に関係なしに同一の制御プログラムを使える。

【0017】

【実施例】以下、本発明の一実施例を図面を参照して説明する。図2はエアコン全体の構成を示すもので、インテークドア10の開度に応じて外気吸入口4または内気吸入口5から吸い込まれた空気は、ブロワモータ1によりエバポレータ2へ送られ冷却される。次に上部エアミクスドア11及び下部エアミクスドア12の開度に応じて、ヒータコア3をバイパスする冷たい空気とヒータコア3を通過することにより再加熱された空気が下流側で混合され、吹出口を切り換えるためのフロアドア13及びベントドア14の開度に従ってデフ吹出口6（フロントガラスへ）、ベント吹出口7（乗員上半身へ）、及びフロア吹出口8（乗員足元へ）から温調風として車室内に吹出され、車室内の温度制御が行われる。

【0018】制御装置20は、デフ吹出口6の吹出空気温度を検出するデフダクトセンサ21、ベント吹出口7の吹出空気温度を検出するベントダクトセンサ22、フロア吹出口8の吹出空気温度を検出するフロアダクトセンサ23、車室外の空気温度を検出する外気温センサ24、車室内の空気温度を検出する車室温センサ25、及び日射の強さを検出する日射センサ27からの検出信号を入力とし、各ドア10～14の開度を調節する電動アクチュエータ15～19とブロワモータ1を制御するものであり、乗員が各種エアコン操作を行うためのスイッチやエアコンの運転状態を乗員に知らせるための表示装置を備えたコントロールパネル26を有している。

【0019】図3はこの制御装置20のより詳細な構成を示すもので、図2と同一符号を付けたものは図2と同一物を示している。この他に、図2では省略したが、コンプレッサをオンオフさせるためのコンプレッサリレー40、エンジン冷却水の流れを開閉制御するための負圧弁41が、制御装置20の負荷として備えられている。

【0020】制御装置20内の1チップマイコン30は、図示を省略した処理装置、ROM、RAM、I/Oポート、A/D変換器等を内蔵している。ドライバ31は電動アクチュエータの回転制御（正転、逆転、停止）を、ファンコントロール回路32は送風用ブロワモータ1の印加電圧の制御を、ソレノイドドライバ33は外部ソレノイドであるコンプレッサリレー40、負圧弁41のオンオフ制御をそれぞれ行う。

【0021】プログラムや制御定数格納用のROM38及びデータを記憶するためのRAM39は、制御装置20の内部で相互に結線され、1チップマイコン30の指令にもとづき動作するが、もしマイコン30内のROM、RAMが十分なメモリ容量を有しておればこれらは不要である。

【0022】制御装置20に組込まれたコントロールパネ

10

20

30

40

50

ル26上の操作スイッチ35からの信号は、インタフェース回路34を介してマイコン30へ入力され、また現在のエアコンの運転状態や設定された目標温度等は、インジケータドライバ36を介してインジケータ37へ出力され、乗員に表示される。

【0023】図4はマイコン30で実行される制御プログラムの例を示すフローチャートで、まずステップ500でマイコン内部のレジスタやRAMの初期設定を行い、エアコンシステムに対する入出力環境を整える。次にステップ501の繰り返し処理により、実際のエアコンシステム10の制御ステップ502～511を電源が切られるまで繰り返し実行する。即ち、ステップ502では、各種センサ21～25、27等から入力された温度信号等を1チップマイコン30内に読み込み、ステップ503では前ステップで読み込んだ温度信号等のデータに対してセンサの非直線性の補正や単位の変換等を行って、内部データに変換する。次にステップ504により、目標室温である制御目標温度 T_{so} の演算を行う。この演算は、乗員の調節操作なしに快適な温度空間が維持できるように、外気温度や運転モードに応じて乗員の選定した設定温度の補正を行う。次のステップ505では、上記制御目標温度 T_{so} と車室温センサ25により検出された実際の車室温 T_r との偏差をはじめ、外気温度 T_a や日射量 Z_m の値に応じて、快適な温度空間を維持するために必要な吹出目標温度 T_{do} の演算を行う。次にステップ506により、エアミクス(A/M)ドア11、12の開度演算を行う。ここでは、前ステップで演算された目標吹出温度 T_{do} と各吹出モードに応じて対応するダクトセンサ21、22、23により検出された実際の吹出温度 T_d との偏差を0に近づけるために必要なエア *

$$\begin{aligned} T_{do} &= f(T_a) - K_z \cdot Z_m + K_r(T_{so} - T_r) \\ &= 40 - K_a \cdot T_a - K_z \cdot Z_m + K_r(T_{so} - T_r) \quad \dots(1) \end{aligned}$$

ここで、 $f(T_a) = 40 - K_a \cdot T_a$ ：基準吹出温度

T_a ：外気温度

Z_m ：日射量

T_{so} ：制御目標室温

T_r ：車室温度

K_a, K_z, K_r ：制御定数

であり、基準吹出温度 $f(T_a)$ は日射がなく、車室温 T_r が制御目標室温 T_{so} に到達している時、乗員が最も快適と感ずる吹出温度で、図5に示すように外気温度 T_a の上昇にともないほぼ直線的に低下する特性を持つ。

【0027】この式(1)の演算を、従来のアセンブリ言語によりプログラミングする場合、各プログラム変数に相当する記憶領域をまずRAM上に定義する。自動車用エアコンの制御には、温度の変化範囲とその分解能の要求により、1つの変数について2バイト程度の記憶領域が必要であり、このため図6(b)に示すように外気温度 T_a 、日射量 Z_m 、車室温 T_r 、制御目標室温 T_{so} 、目標吹出温度 T_{do} それぞれに対し2バイトの記憶領域を定義する。ここで算出しようとする温度 T_{do} 以外の変数の

* ミクスドア11、12の開度が演算される。

【0024】以上で温度制御関係の演算は終了し、次にステップ507により吹出風量を決定するためのブロワモータ制御目標電圧が演算される。ここでは一般に制御目標温度 T_{so} と実際の車室温 T_r との偏差がほぼ0の場合低風量に、またこの偏差が大きくなるに従って高風量になるよう演算される。次のステップ508では、外気温度 T_a 、日射量 Z_m 、吹出温度 T_d 等に応じて最適な各吹出口6～8の風量分配や熱負荷に応じた内気循環空氣の決定を行う。次のステップ509によりコンプレッサのオンオフの判断を実施する。一般には熱負荷の少なくなる低外気温条件で強制的にコンプレッサをオフとするよう判断される(リレー40動作)。またステップ510では、熱源であるエンジン冷却水の流れを開閉するためのウォーターコックの開閉(負圧弁の開閉)の判断を行う。最後にステップ511により、以上演算された各アクチュエータ類への制御出力信号がまとめて出力され、それぞれエアミクスドアの開度、ブロワモータ印加電圧、吸込口、吹出口のドア状態、コンプレッサ、ウォーターコックのオンオフ、及びコントロールパネル上のインジケータ類の制御が行われる。以上のステップ502～511が繰り返し実行されることにより、乗員が設定した条件に応じて車室内が快適な状態に維持される。

【0025】ここで図4のステップ505および509を例にとり、本発明の特徴とするプログラムの例を説明する。まずステップ505の目標吹出温度 T_{do} の演算式の一例を以下に示す。

【0026】

【数1】

値は、ステップ503および504ですでに計算され、定義領域に値が記憶されているので、これらの値を用いて式(1)の演算を図6(a)に示すフローチャートで実行する。即ちステップ600にて基準吹出温度 $40 - K_a \cdot T_a$ を演算し、結果を T_{do} の定義領域へ一時的に格納する。ここで図中のA→BはAを求めてBへ代入することを意味する。次にステップ601にて、ステップ600で求めた T_{do} の値から日射量 Z_m に対する吹出目標温度の補正量 $K_z \cdot Z_m$ を差し引き、その結果を再度 T_{do} の領域へ格納する。次にステップ602にて、さらに室温偏差に基づく目標吹出温度の補正值である $K_r(T_{so} - T_r)$ をステップ601で求めた T_{do} に加え、その結果を再び T_{do} の領域に格納する。以上の手順を実行することにより T_{do} の定義領域に、前述の式(1)で計算される目標吹出温度データ T_{do} が得られる。ところがこれらの各ステップでは、2バイト長データ同志の乗算が必要で、従来の8ビットマイコンの場合は各ステップで倍長乗算用のサブルーチンを用いる必要がある。この2倍長乗算の方法は周知のものであるのでその詳細は省略するが、1バイト長データの乗

算回数4回と、それらの結果を適当に桁合わせして行う加算とが必要であり、また乗算結果が4バイト長となるから、それに見合ったメモリ領域を用意する。

【0028】以上の式(1)の処理をアセンブリ言語で記述したプログラムリストの例を図7(a)(b)に示す。(a)は式(1)を計算する図6(a)のプログラム、(b)は其中で用いる2バイト長乗算のサブルーチンである。本リストは(株)日立製作所製の8ビット1チップマイコンHD6801を使った場合の例であるが、これは合計約50行から成っている。またプログラムの内容はリストを見ただけでは分かりにくく、その演算仕様の一部を間違いなしに変更するには、多くの時間を必要とする。さらにこの例で示したマイコンHD6801は2バイトデータ同士の加減算機能を備えているので、比較的単純にプログラムが組めるが、自動車用エアコン制御用に使用されることが多い同社製のHD6805の場合は2バイトデータ同士の加減算機能がないため、プログラムはさらに長く、複雑になる。

【0029】本発明では、このようなアセンブリ言語の持つ問題点を解決するために、式(1)の演算を高級言語で記述する。図1はマイコン制御用言語として多く使用されているC言語で式(1)を記述した例を示すものである。同図から明らかなように、プログラムは極めて簡単になり、かつプログラムの記述そのものが式(1)とほぼ同一であり、プログラムの作成や演算仕様の変更がアセンブリ言語と比較して極端に簡単になる。

【0030】次に、図4におけるステップ509のコンプレッサオンオフの判定プログラムの比較について説明する。図8はコンプレッサのオンオフ判定特性の一例である。前述のごとくコンプレッサは低外気温では作動させる必要がなく、また低熱負荷でのコンプレッサ運転によるコンプレッサの損傷を防止する目的からも、一般的に外気温 T_a が 0°C 付近以下では、コンプレッサを強制的にオフさせる。そして外気温センサからの信号の微小な変動の影響による切替点でのチャタリングを防止するために、外気温が $\#Hys$ (図8では 1°C)以上でコンプレッサオン、外気温が $-\#Hys$ 以下ではコンプレッサオフとなるように、ヒステリシスを設けたオンオフ判定を行う。

【0031】図9及び図10は図4のステップ509の処理のためのフローチャートで、まず、ヒステリシス特性を持った大小判定のサブルーチン(これは汎用性があるのでサブルーチンとした)を図10にて説明する。図10(b)はメモリ上のFlagエリアを示しており、このLSBの1ビットを、コンプレッサのオン/オフ状態を記憶するための B_{comp} として用いる。また、図10(a)内の $[IX]$ は、インデックスレジスタIXに格納されているデータの値を意味し、これはヒステリシス付きの大小判定が行われるデータ、つまり外気温 T_a である。図10(a)のステップ900では、まず現在コンプレッサが

オン状態にあるか否かを判定する。もし $B_{comp}=$

“1”、即ち現在コンプレッサがオン状態であれば、ステップ901で $[IX]$ を $T_a + \#Hys$ に置き換え、もし $B_{comp}=$ “0”、即ちオフ状態であれば、ステップ902で $[IX]$ を $T_a - \#Hys$ に置き換える。つづくステップ903では $[IX]$ の正負を判定する。これが正のときは、もし今まで $B_{comp}=$ “0”ならば $T_a - \#Hys > 0$ を、今まで $B_{comp}=$ “1”ならば $T_a + \#Hys > 0$ を意味するが、これは図8から明らかなように、いずれのときもコンプレッサをオンとする状態であるから、ステップ904で $B_{comp}=$ “1”とする。逆にステップ903の判定が負のときは、もし今まで $B_{comp}=$ “0”なら $T_a - \#Hys < 0$ を、もし今まで $B_{comp}=$ “1”なら $T_a + \#Hys < 0$ を意味し、これはいずれもコンプレッサをオフとすべき状態であるから、ステップ905で $B_{comp}=$ “0”とする。

【0032】図9は図4ステップ509を実行する主プログラムである。図10(a)のステップ901または902にて $[IX]$ は必ず書き換えられるから、外気温データ T_a の値は図10(a)の処理を開始するごとにレジスタIXにセットする必要がある。このため、図9(b)のように、メモリ上にWorkというワーキングエリアを T_a 格納用のエリアとは別に用意する。そして図9(a)のステップ800にて、外気温 T_a の値をWorkエリアにコピーし、さらにステップ801にてインデックスレジスタにWorkエリアのアドレス、即ち $\#Work$ を代入したうえで、ステップ802でサブルーチンと呼ぶ。

【0033】以上の図9及び図10の処理をアセンブリ言語で記述したリストをそれぞれ図11(a)及び(b)に示す。前述の図7(a)(b)と比較するとプログラムステップ数は大変少ないが、やはりこのリストのみから何の処理を行っているプログラムかを読み取ることは非常に難しい。

【0034】図12は、上記の判定処理509をC言語で記述した場合のプログラム例であり、プログラム行数で見れば従来の図11(a)(b)と比べて減らないが、その内容はC言語を扱っている技術者であれば、容易に理解可能で、特性の変更も容易に行える。

【0035】以上の図1、図12の実施例で説明したごとく、高級言語であるC言語を使って制御プログラムを記述することにより、プログラム記述行数が減少し(特に数値演算の処理が減る)、内容の理解が容易となり、プログラムの仕様変更や作動内容の検証が容易にかつ確実に行えるようになるという効果がある。またC言語のような代表的な高級言語は、殆どの汎用マイコンに対して良質なコンパイラがすでに供給されており、各マイコンの構造が異なってもそれぞれ専用のコンパイラを搭載しておけば、制御プログラムは同一のものが使用できる。

【0036】一方、翻訳されたオブジェクトプログラム

10

20

30

40

50

(機械語)を格納するエリアが必要となり、またオブジェクトプログラムはコンパイラによる自動翻訳のために多少効率が悪い命令列となることが避けられず、処理速度がアセンブリ言語により作成されたプログラムよりも通常は低下する。しかしこれらのメモリ容量の増大や処理速度の低下は、前述のように、ハードウェアの進歩により十分吸収可能である。

【0037】次に、本発明の他の実施例として、演算式(1)の計算および図8に示したプロセスのオンオフ判断処理を、高級言語BASICで記述した例をそれぞれ図13及び図14に示す。C言語を用いた図1及び図2と比較すると、プログラムの行数、理解しやすさのいずれもほぼ同等であり、C言語を用いた場合と同様の効果がある。また、BASIC言語はパーソナルコンピュータ用の言語として広く浸透しており、C言語と比較してプログラムの作成や変更を行える人間の範囲がさらに広がる。同時にBASIC言語はインタプリタ言語と呼ばれ、プログラム実行環境状態であっても、一時的に停止させてプログラムの変更、追加を容易に行えるため、プログラム仕様開発途上の仕様修正にも即座に対応できるという効果がある。

【0038】一方BASIC言語は一般に処理速度が遅いが、マイコンの処理能力の向上により処理速度の問題は解消しつつある。その内部にBASICプログラムを直接実行するためのBASICインタプリタを内蔵したものが発表されており、必要となるROM容量や処理速度はさらに改善される。その他、BASICの中には、中間コード(Iコード)インタプリタ方式のものや、C言語と同様にコンパイラ方式のものもあり、これらを使用することによって、製品としてのプログラムのコンパクト化やプログラム処理の高速化が可能となる。

【0039】その他の高級言語も同じように応用可能であり、また自動車制御やエアコン制御に限定したより高速でコンパクトな専用的高级言語を設計し、用いることもできる。

【0040】次に本発明の他の実施例として、プログラム開発装置と自動車用エアコンを結びつけ、プログラム開発装置上のアセンブリ言語あるいは高級言語で記述されたプログラムによりエアコンの制御を行うようにした実施例を図15に示す。ここでプログラム開発装置というのは、プログラムの作成、修正等に用いるエディタ、作成した高級言語プログラムを翻訳するコンパイラ、アセンブリ言語プログラムを翻訳するアセンブラ、チェックや評価用のデバッグ及びエミュレータ等の機能を持ち、1チップマイコン用のOSや制御プログラムの作成、修正等を行うものである。

【0041】図15の上側のブロックは図2のエアコンの制御装置20の内部構成例を示しており、図3と異なるのは、図3から1チップマイコン30を除去し、その代わりに各入出力端子へのインターフェース回路やドライバ

回路からの信号線がバスライン43を介して引き出されていること、各センサ23~25、27入力部にインターフェース42が設けられたことである。

【0042】一方、図15下側のブロックはプログラム開発装置59の内部構成例を示しており、マイコン45、ROM46、RAM47、タイマカウンタ48、記憶装置50、51(フロッピーディスク、ハードディスク、光ディスク、バブルメモリ等)、キーボード53、ディスプレイ装置55、外部に接続された汎用の計測装置57、処理装置58(プリンタ、プロッタ、データロガー、EPROMライタ等)、インターフェース回路49、52、54、プログラム開発装置と外部の汎用装置を接続するための汎用インターフェース回路56(RS-232C等)、バス60から成っている。バス60と制御装置20のバス43は、コネクタ44で接続されている。

【0043】以上の構成に示されているように、プログラム開発装置59は、リアルタイムマルチタスクOSを搭載し、大容量のRAM47や大容量高速の記憶装置50、51を備え、リアルタイムOS上でプログラム開発に必要なエディタ、各種アセンブラ、C言語やBASIC言語をはじめとした高級言語のコンパイラ、インタプリタ等の動作が可能であり、引き出されたデータバス60、43を介して制御装置20をマイコン45より直接操作でき、その上プログラム開発装置59は、AC100Vまたは自動車用DC電源でも動作可能とする。

【0044】従って、このプログラム開発装置を使用してアセンブリ言語あるいは高級言語によりエアコン用制御プログラムを作成し、そのプログラムをマイコン45上で走らせ、制御装置20を介して車両のエアコンを直接制御できる。特に実車における快適性評価の走行テストにおいては、本システム全体を車載し、プログラム開発装置59により空気制御を実行するとともに、マルチタスク機能を利用して各センサから入力された読み取りデータ、制御プログラム内部の演算データ、各アクチュエータの動作状況、また汎用の外部装置を介して読み取られた車両各部の温度、圧力データ等をリアルタイムでディスク等の記憶装置50、51に順次記録し、後の分析に備えるとともに、ディスプレイ装置55や計測装置57に各データの変化経過をリアルタイムで表示させ、必要に応じて車両走行中でもエアコン制御プログラムの修正や再コンパイルを可能とする。なお、プログラム開発装置59を使用する場合には、その機種により制御装置20に一体に設置されているコントロールパネル26は不要である。

【0045】また最終製品となる図3の制御装置には、本実施例のプログラム開発装置で使用したOSから、製品に不要な機能のみ削除した共通OSを搭載すればよく、他の変更なしで動作検証済みの制御プログラムが得られる。

【0046】プログラム開発装置のOSとして使用可能なものは、現在汎用品のうちの代表的なものとしてのO

S9、I-TRON、μ-TRON、各マイコンメーカーが供給する各リアルタイムOS等があるが、本システム専用のコンパクトな高性能OSを開発し、使用することも可能である。

【0047】本実施例によれば、制御プログラム開発、仕様変更期間の大幅短縮と、同プログラムの品質を大幅に向上できるという効果がある。

【0048】次に信号やデータの入出力を行うにあたり、プログラムから直接入出力インターフェースアドレスへの制御を行わず、OS上のシステムコール動作により入出力を実現した実施例を図16に示す。これはOS上で動作するC言語から演算結果を出力するプログラム例であり、図12のプログラムにより演算されたコンプレッサオンオフを示す Bcompを含んだ1バイトの制御出力を、Oportと定義された実際の出力ポートに出力する。

【0049】本実施例によれば、制御回路の構成、あるいは使用部品の相違により入出力ポートのアドレスや入出力手順が変わっても、これらの相違はすべてOSの中で吸収することができるため、図16に示した入出力処理プログラムも含めて、制御プログラムは一切その影響を受けることがない。従って他機種用に制御プログラムを流用する場合や、図15に示したプログラム開発装置上で作成した制御プログラムを、入出力アドレス等が異なる図3の装置に適用する場合にも、制御プログラムは全く変更せず、そのまま流用でき、制御プログラムの開発期間の大幅短縮と、同プログラムの品質を大幅に向上できるという効果がある。

【0050】以上、各種の実施例を自動車用エアコンの制御について示したが、エンジン制御等、1チップマイコンの利用は数多くあり、それらにも本発明が利用可能なことはいうまでもない。

【0051】

【発明の効果】本発明によれば、高級言語はプログラム作成が容易かつ内容も分かり易いので、特に訓練されたプログラマ以外でも制御プログラムの作成、変更や検証を容易に行え、また機種対応のコンパイラによりマイコンの構造が隠されるから、マイコンの機種や使用部品が異なっても同一の制御プログラムを利用できるという利点がある。このため、自動車制御装置の開発工数やマンパワーが少なくなり、製造コストの低減が図れる。

【0052】また、プログラム開発装置と制御装置のOSを共通仕様とすることで、製品の開発期間中において実験途上または屋外車上の任意の場所での制御プログラ

ムの修正、変更が行え、作成された制御プログラムは大きな変更なしにそのまま制御装置で使用できる効果があり、また製品の開発期間中にマイコン内部の任意データを簡単に記録、表示でき、それらのデータの分析（グラフ化等の処理）も自動的に行える。さらに制御プログラムの入出力処理にOSへのシステムコールを用いることで、入出力に関するアドレスの違い等をOSで吸収できるので、制御プログラムの互換性を一層高めることができ、上記の諸効果と合わせて制御プログラムの開発、仕様変更期間の大幅短縮と同プログラムの信頼性の向上に大きく寄与できるという効果がある。

【図面の簡単な説明】

【図1】本発明の特徴とする制御プログラムの一例を示す図

【図2】自動車用エアコンとその制御装置の全体構成を示す図

【図3】制御装置の内部構成例を示したブロック図

【図4】自動車用エアコンの制御手順の概要を示すフローチャート

【図5】基準吹出温度の特性例を示す図

【図6】目標吹出温度の演算手順を示す図

【図7】図6に示す手順をアセンブリ言語でプログラム化した例を示す図

【図8】コンプレッサのオンオフ特性の例を示す図

【図9】コンプレッサオンオフの判定手順を示すフローチャート

【図10】コンプレッサオンオフの判定手順を示すフローチャート

【図11】図10、11に示す判定手順をアセンブリ言語でプログラム化した例を示す図

【図12】図10、11に示す判定手順をC言語でプログラム化した例を示す図

【図13】図1に対応するプログラムをBASIC言語で示した例を示す図

【図14】図12に対応するプログラムをBASIC言語で示した例を示す図

【図15】本発明の特徴とするプログラム開発装置の一実施例を示す図

【図16】OSのシステムコールにより入出力を行うプログラムの例を示す図である。

【符号の説明】

20…制御装置、30…1チップマイコン、43、60…バス、44…コネクタ、45…マイコン、46…ROM、47…RAM、59…プログラム開発装置。

【図13】

[図 1 3]

$$T_{do}=40, -K_a * T_a - K_z * Z_m + K_r \quad (T_s - T_r)$$

【図1】

[図 1]

```

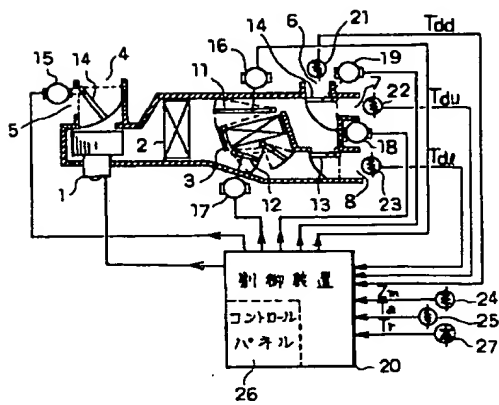
tdo=40.+ka*ta-kz*zm+kr*(ts-tr);
return (tdo);

```

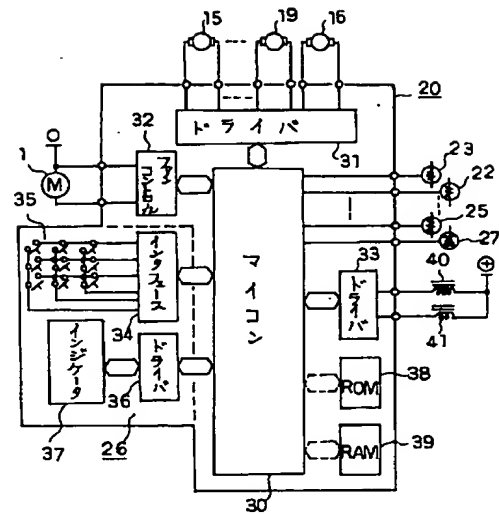
【図2】

【図3】

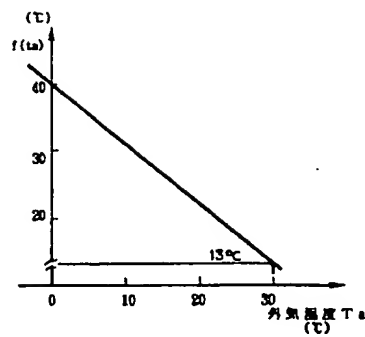
[図 2]



[図 3]

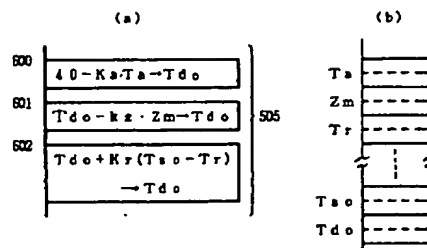


【図5】



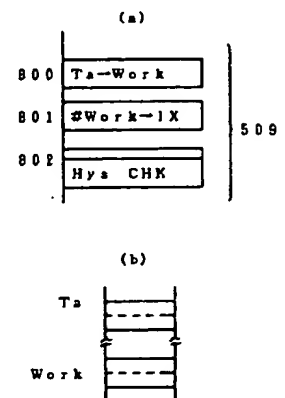
【図6】

[図 6]



【図9】

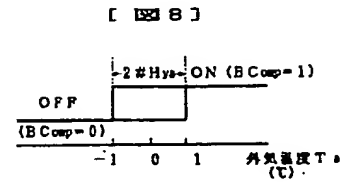
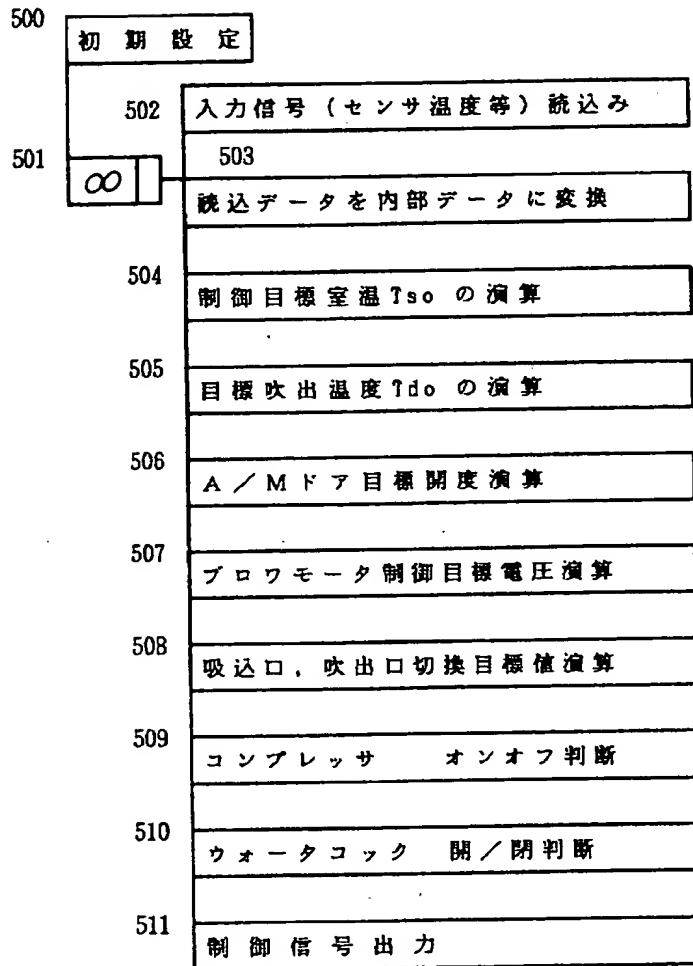
[図 9]



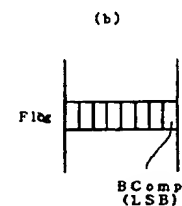
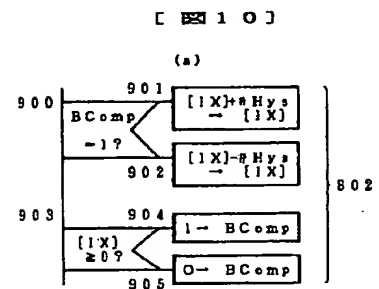
【図4】

【図8】

【図4】



【図10】



【図12】

【図12】

```

if (flag & bcomp)
    ta += Hys;
elss
    ta -= Hys;
if (ta >= 0.)
    flag |= bcomp;
elss
    flag &= ~bcomp;

```

【図16】

【図16】

```

fd = open ("oport", 1);
write(fd, *flag, 1);
close(fd);

```

【図7】

【図11】

[図 7]

[図 1 1]

```

(a)
OprTdo  ldd #ka
        std Param1
        ldd Ts
        std Param2
        bsr Mul10
        ldd #40
        subd MulAns+1
        std Tdo

        ldd #kz
        std Param1
        ldd Zn
        std Param2
        bsr Mul10
        ldd Tdo
        subd MulAns+1
        std Tdo

        ldd #kr
        std Param1
        ldd Ts
        std Param2
        bsr Mul10
        ldd Tdo
        addd MulAns+1
        std Tdo
        rts

```

```

(b)
Mul 10  clr MulAns
        clr MulAns+1

        lda Param1+1
        ldb Param2+1
        mul
        std MulAns+2

        lda Param1+1
        ldb Param2
        bsr MulSub10

        lda Param1
        ldb Param2+1
        bsr MulSub10

        lda Param1
        ldb Param2
        mul
        addd MulAns
        std MulAns
        rts

MulSub10 mul
        addd MulAns+1
        std MulAns+1
        lda #0
        adca MulAns
        sat MulAns
        rts

```

```

(a)
Contcomp ldd Ta
        std Work
        ldx #Work
        bsr HysChk
        lda Flag
        sta OPort
        rts

```

```

(b)
* List 5
HysChk   lda Flag
        bita #BComp
        bne HysChk10
        ldd 0,X
        addd #Hys
        bra HysChk20

HysChk10 ldd 0,X
        subd #Hys
HysChk20 bPl HysChk30
        lda Flag
        anda #~BComp
        bra HysChk40

HysChk30 lda Flag
        ora #~BComp
HysChk40 sta Flag
        rts

```

【図14】

[図 1 4]

```

IF LAND (Flag, BComp) THEN
    Ta=Ta+Hys
ELSE
    Ta=Ta-Hys
ENDIF
IF Ta>=0. THEN
    Flag=LOR (Flag, BComp)
ELSE
    Flag=LAND (Flag, LNOT (BComp))
ENDIF

```

[15]

